

# Ozobot Maze Challenge

**Year level band:** 3-4 (can also be adapted for 5-6 as a first lesson with Ozobot and Ozoblockly)

**Description:** Students are introduced to Ozoblockly and basic programming concepts. Using Ozoblockly, students program Ozobot to follow a path and travel through a maze that they have created.

**Resources:**

- Ozobots (1 per group of 2 or 3)
- Tablets or computers to run Ozoblockly website (<http://ozoblockly.com>) and Ozobot Games website (<http://games.ozoblockly.com>)
- Paper and pens
- (For creating a 3D maze:
  - o Craft materials e.g. paper, wood, cardboard, boxes, tubes
  - o Lego bricks or similar)

**Prior Student Learning:**

**Digital Technologies:** Students have done some work on algorithms and used some basic visual programming.

**Maths:** Students understand right angles.

By the end of Year 4, students record simple solutions to problems through text and diagrams and develop their designing skills from initially following prepared algorithms to describing their own that support branching (choice of options) and user input. Their solutions are implemented using appropriate software including visual programming languages that use graphical elements rather than text instructions.

Year	Content Descriptors
3-4	Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them (ACTDIP010) (Digital Technologies, 3-4)
Year 4 Maths	Compare angles and classify them as equal to, greater than, or less than, a right <a href="#">angle</a> (ACMMG089)

Element	Summary of tasks
Learning hook	Introduce Ozobot if not used before.



	<p>(If Ozobot colour codes have been used before explain that in this activity, we will not be controlling Ozobot by drawing lines and colour codes.)</p> <p>Have one or two Ozobots preloaded with simple programs e.g. one of the levels from the Shape Tracer games (<a href="http://games.ozoblockly.com/shapetracer-basic">http://games.ozoblockly.com/shapetracer-basic</a>)</p> <p>Show that Ozobot can move on its own without the need to draw lines or colour codes. Play your preloaded program.</p> <p>How do you think we can control Ozobot?</p> <p>In this activity, we will write a program on the computer and load the instructions onto Ozobot.</p>
<p>Achievement Standards</p> <p>Learning Map (Sequence)</p>	<p>Students define simple problems, design and implement digital solutions using algorithms that involve decision-making and user input.</p> <ul style="list-style-type: none"> <li>• Students describe the sequence of moves that Ozobot needs to make.</li> <li>• Students use Ozoblockly to write a sequence of steps that navigate Ozobot along a path.</li> <li>• Students can program Ozobot to move along a path that they have designed</li> <li>• Students can debug their algorithms and programs</li> </ul>
<p>Learning input</p>	<p>Show a simple example of Ozoblockly e.g. one of the levels from the Shape Tracer games (<a href="http://games.ozoblockly.com/shapetracer-basic">http://games.ozoblockly.com/shapetracer-basic</a>)</p> <div data-bbox="459 1019 1375 1328" style="border: 1px solid blue; padding: 5px;"> </div> <p>Teacher demonstrates both the 'virtual Ozobot' and loads the program onto Ozobot and shows this following the same path.</p> <p>Sample questions to ask the class:</p> <ul style="list-style-type: none"> <li>• What would happen if I changed the light colour?</li> <li>• What would happen if I changed the speed?</li> <li>• What would happen if I choose 'slight left' instead of 'left'?</li> </ul>
<p>Learning construction</p>	<p>Students understand that Ozobots can be programmed using Ozoblockly.</p> <p><b>Activity 1</b></p> <p>They work in groups of 2 or 3 to work through a few of the Shape Tracer activities: <a href="http://games.ozoblockly.com/shapetracer-basic">http://games.ozoblockly.com/shapetracer-basic</a></p>



	<p>They should write their programs in Ozoblockly and run with the 'virtual Ozobot' and debug if necessary, before loading onto Ozobot.</p> <p>As students work, teacher asks questions:</p> <ul style="list-style-type: none"> <li>• What distance is one step on the real Ozobot?</li> <li>• What angle does Ozobot turn when it is programmed it to turn right?</li> </ul> <p>(right turn is approximately a right angle of 90 degrees, slightly right turn is approximately 45 degrees)</p> <p><b>Activity 2</b></p> <p>Students design a simple path or maze and program Ozobot to follow the path or solve the maze.</p> <p>Using <a href="http://ozoblockly.com">http://ozoblockly.com</a> rather than the games website, gives more blocks – students can explore the additional blocks available.</p> <p>Depending on time, the maze could be simply drawn on paper or you could use craft materials or lego to build the maze.</p>
Learning demo	<p>Once students have designed their mazes and successfully loaded programs, have students try to solve each other's' mazes.</p> <ul style="list-style-type: none"> <li>• What worked well and did not work well?</li> <li>• What did you discover about the different speed options?</li> <li>• What do we know about the distance of the steps?</li> <li>• Did you change anything?</li> </ul>
Learning reflection	<p>Bring together the observations from the learning demo and, with the students, come up with a list of findings, e.g.</p> <ul style="list-style-type: none"> <li>• What angle is a turn?</li> <li>• What distance is a step?</li> <li>• Does Ozobot behave better at a certain speed? When would you want to use slow, medium and fast?</li> </ul> <p>Consider ways to make a more complicated maze – e.g. with tunnels and ramps – how would we need to alter the programs?</p>

**Assessment:**

Formative Assessment:

- Teachers observe students using the Ozobots, creating their algorithms and debugging their programs.
- Use questioning to elicit student understanding of the functions of the Ozobot and how their online program translates to the behaviour of the Ozobot.
- You might take photos of the students' work to document their progress, or record the Ozobot in their final demonstration.

	<b>Quantity of knowledge</b>	<b>Quality of understanding</b>
--	------------------------------	---------------------------------



Criteria	Pre-structural	Uni-structural	Multi-structural	Relational	Extended abstract
Algorithms Programming	No visual program written	Algorithm only shows a limited number of instructions which are not linked	Algorithm has enough instructions to complete the task but not linked or not linked in the correct sequence – or the Ozobot only completes part of the maze	Algorithm has instructions linked in the correct sequence to achieve the task – Ozobot can follow a path as designed	Algorithm brings in prior learning and/or independent learning beyond the task and possibly includes additional blocks and features (e.g. loops)
Vocabulary	When describing algorithm, no specific vocabulary is used	The terms instruction or code may be used as a general description	The term algorithm is used as a general description	The terms algorithm and program is used confidently with specific reference to learner's work	Specific vocabulary like decisions and repetition is used, going beyond the set language

## Teacher/Student Instructions:

It is useful for teachers to have read the Teacher Guide prior to using Ozobots with students: <http://files.ozobot.com/stem-education/ozobot-teachers-guide.pdf>

## CSER Professional Learning:

This lesson plan corresponds to professional learning in the following CSER Digital Technologies MOOCs:

F-6 Digital Technologies: Foundations

- Unit 7: Algorithms and Programming
- Unit 8: Visual Programming

F-6 Digital Technologies: Extended

- Unit 2: Algorithms & Programming

See: <http://csermoocs.adelaide.edu.au/moocs>

## Further Resources:

Ozobot Lesson Library (new lessons added monthly): <http://portal.ozobot.com/lessons>



Digital Technologies Hub: [www.digitaltechnologieshub.edu.au](http://www.digitaltechnologieshub.edu.au)

CSER: <https://csermoocs.adelaide.edu.au>



Author: Steven Payne

This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/). Computer Science Education Research (CSER) Group, The University of Adelaide.



CSER Group, The University of Adelaide